
cloudthat
move up.


ANSIBLE

About CloudThat

9+
Years in Business

350k+
Professional Trained

100+
Corporates Trained

100+
Projects Delivered

350+
Cloud Certifications

Leader in Training and Consulting on Cloud, Security, AI/ML, IoT and DevOps

Trained over 350k+ professionals across technologies and geographies

Proven track record of training delivery for all stages of employee lifecycle

Strong team of 125+ certified cloud experts with industry consulting experience

Robust consulting division brings industry prospective to training delivery

Gold Microsoft Partner
Microsoft

Microsoft Partner
2020 Partner of the Year Finalist
Learning Award

aws partner network
Advanced Consulting Partner

Google Cloud Partner

cloudthat

Trainer Profile

VENKATA SUBBARAO B

Cloudthat Consultant

UI | UX | JAVA Full Stack | Architect | Cloud | DevSecOps | SRE | DS

Cloud & Other Platforms
AWS, AZURE, GCP, Solution Architect, DevSecOps, SRE

Front End
ES.Next, JavaScript, React, React Native, Vue, Angular Ext JS, UXD, UI , HTML, CSS, Bootstrap, Tailwind CSS, SASS, Wire Framing , Low & High Fiddle Design, RWD, PWA …

Back End
JAVA, Spring Boot, Node, ExpressJS, PHP, Python, SQL(MySQL, PgSQL), NoSQL (Mongo, Firebase, Elasticsearch, Cassandra, Couchbase), API

Miscellaneous Tools
Docker, Kubernetes, Jenkins, Kafka, Ansible, Terraform, SonarQube, Selenium, Atlassian Stack, Grafana, GraphQL, ELK Stack, Git, SVN, CVS, Redux, MobX, Thunk, Saga, Figma, Adobe XD, Karma & Jasmine, Protractor, Jest, SonarQube, Photoshop, Illustrator, Digital Marketing, Agile Scrum …

cloudthat

JUNIPER NETWORKS

msp CONCEPTS IT Solutions - Execute Your Concepts

Tech Mahindra

UNIVATIONS

IBM | COGNIZANT | CRIO.DO | JIGSAW ACADEMY | TELSTRA | UNEXT | BAJAJ FINSERVE

MANIPAL PRO LEARN | ETEAMINC | PROBITS | MANTT | AWIGN | MYYESM

RAKUTEN | ANTHEM | UST-GLOBAL | MAZENET | TATA SED | HONDA USA | COLORS SOFT

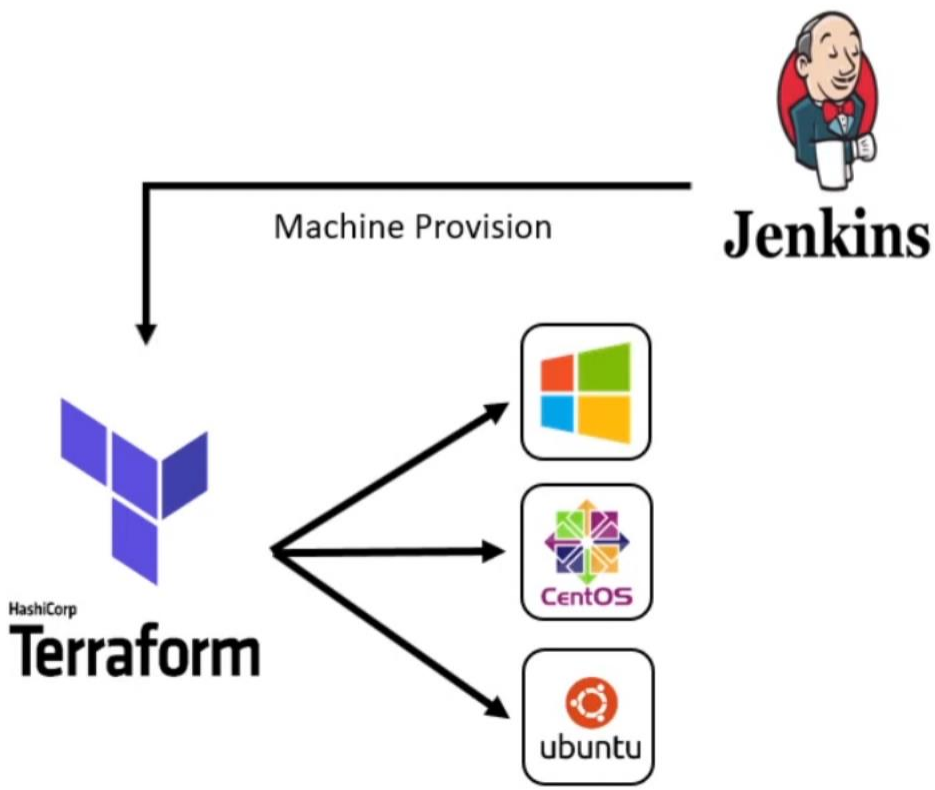
TRAINING CURVE | 6BENCHES | EDUPINNACLE | AIMS MBA COLLEGE | VERZEO

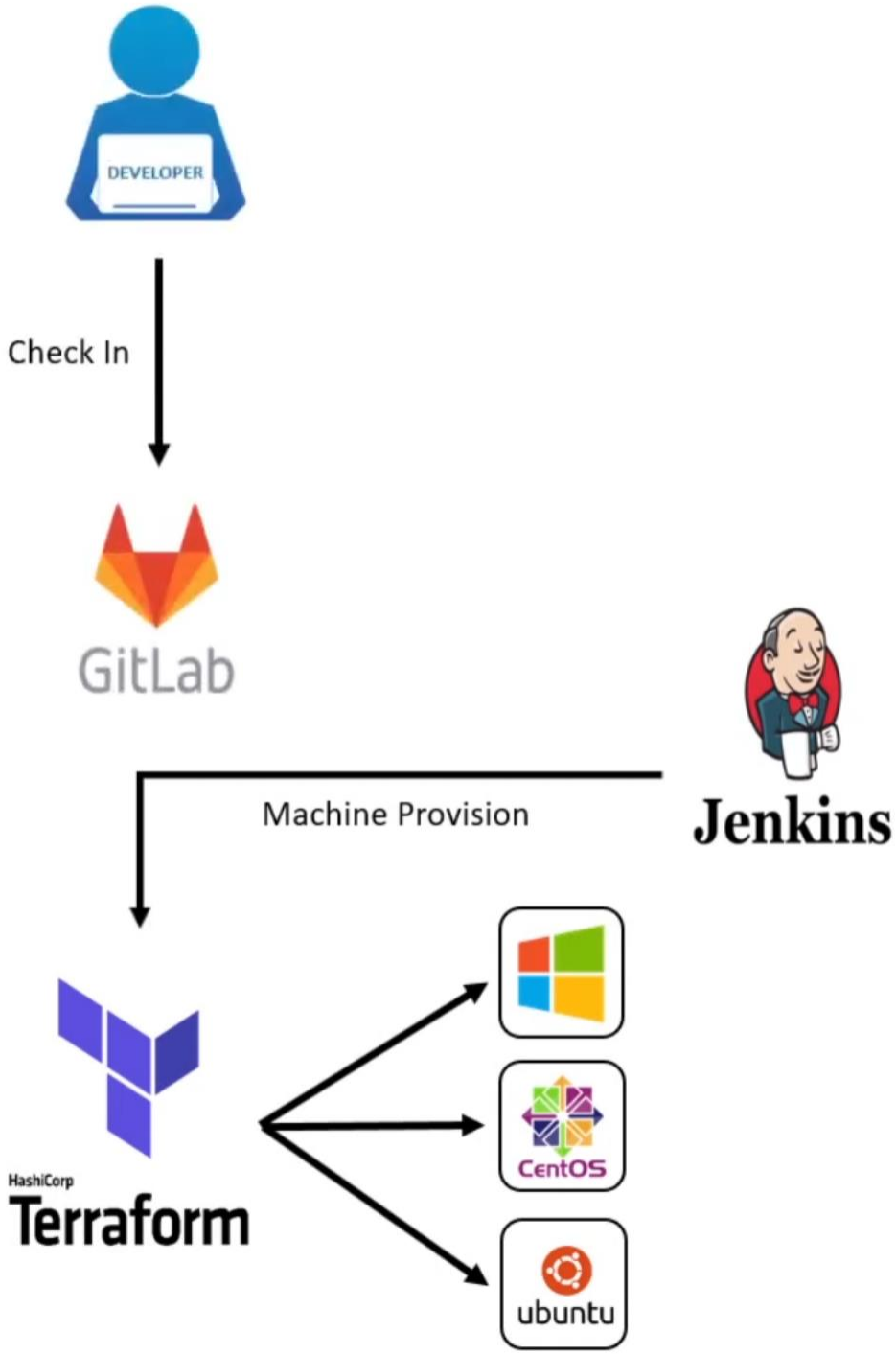
INVENTATEQ | CODING SUPER STAR | VSN TRADERS | IMPACT FOUNDATION | CRAYONZE

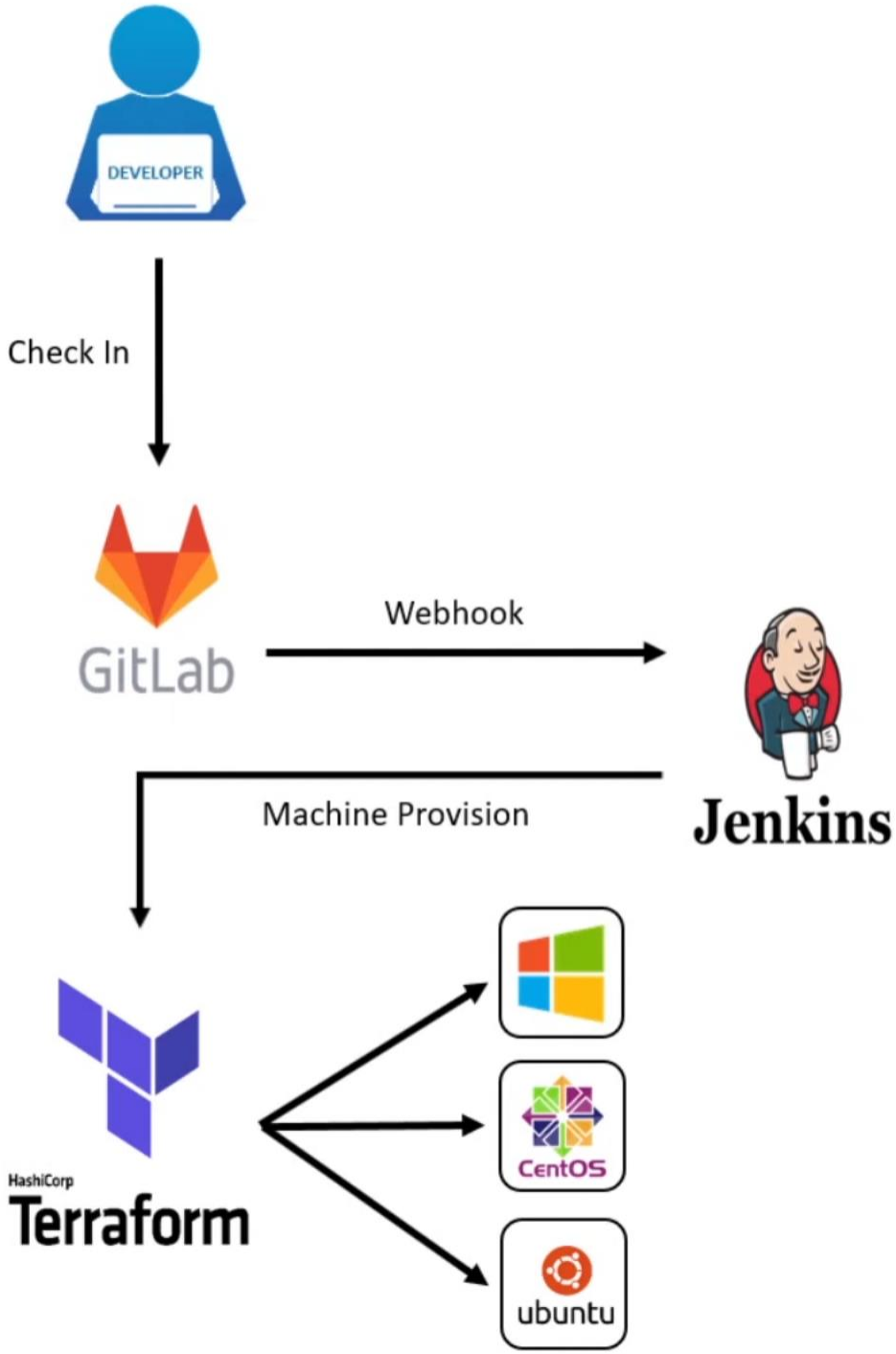
AGENDA:

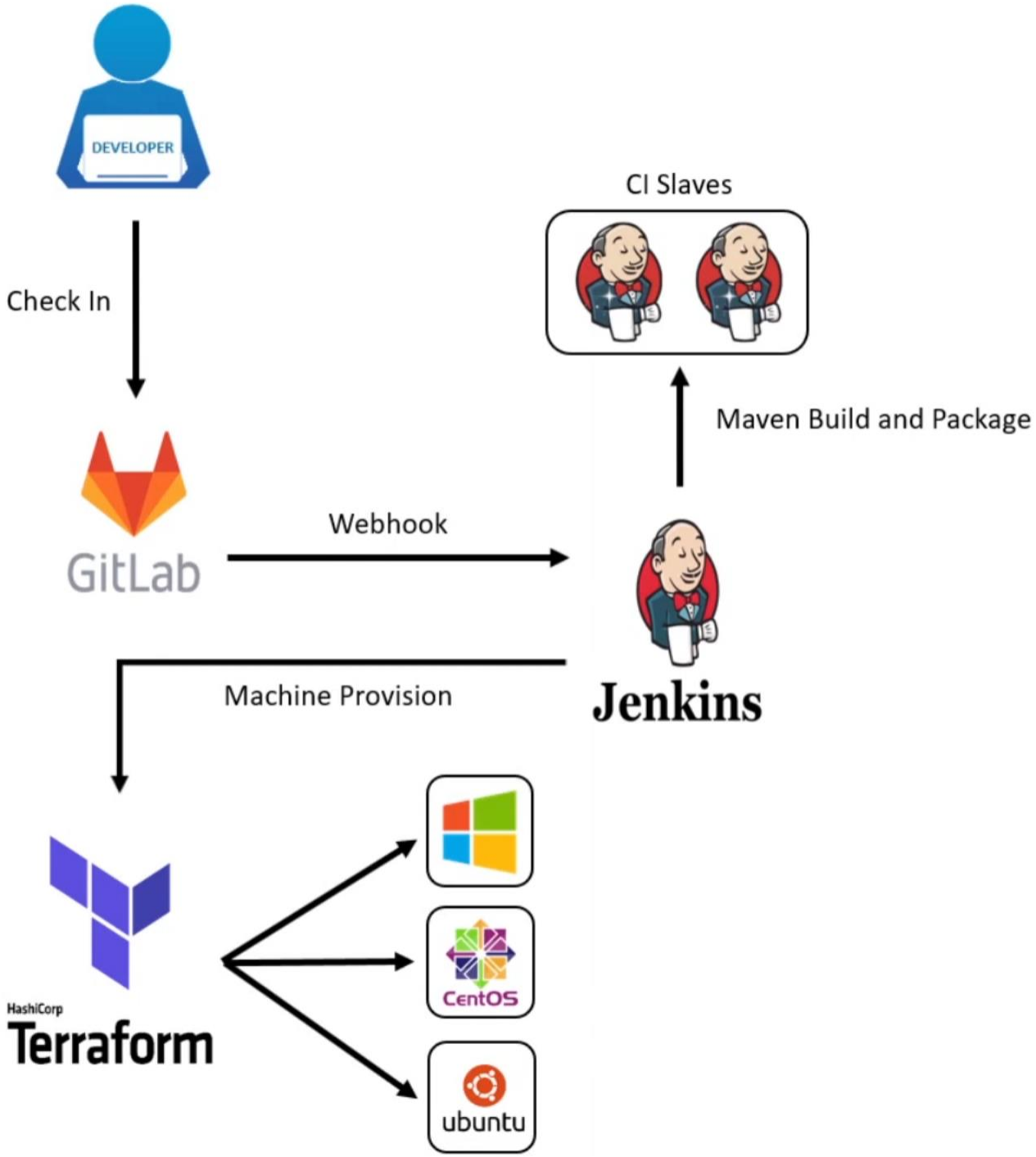
- Managing Variables
- Managing Facts
- Implementing Modules
- Managing Inclusions
- Constructing Flow Control
- Implementing Handlers
- Automation with Ansible
- Implementing Tags
- Handling Errors
- Learn about retrieving data from external sources using Lookups.
- Share work with Ansible Community using Ansible Galaxy
- Optimizing Ansible
- Configuring Connection Types
- Configuring Delegation Configuring Parallelism
- Develop Ansible Playbooks for advanced use cases
- Nested Playbooks
- Managing Dynamic Inventory
- Use Dynamic Inventory in playbooks
- Need Design discussions on Below Topics based on the practical need we can project in Orchestration Area.
- Creating Playbooks for Kubernetes Cluster Deployment using Ansible
- Creating Playbooks for Container Orchestration

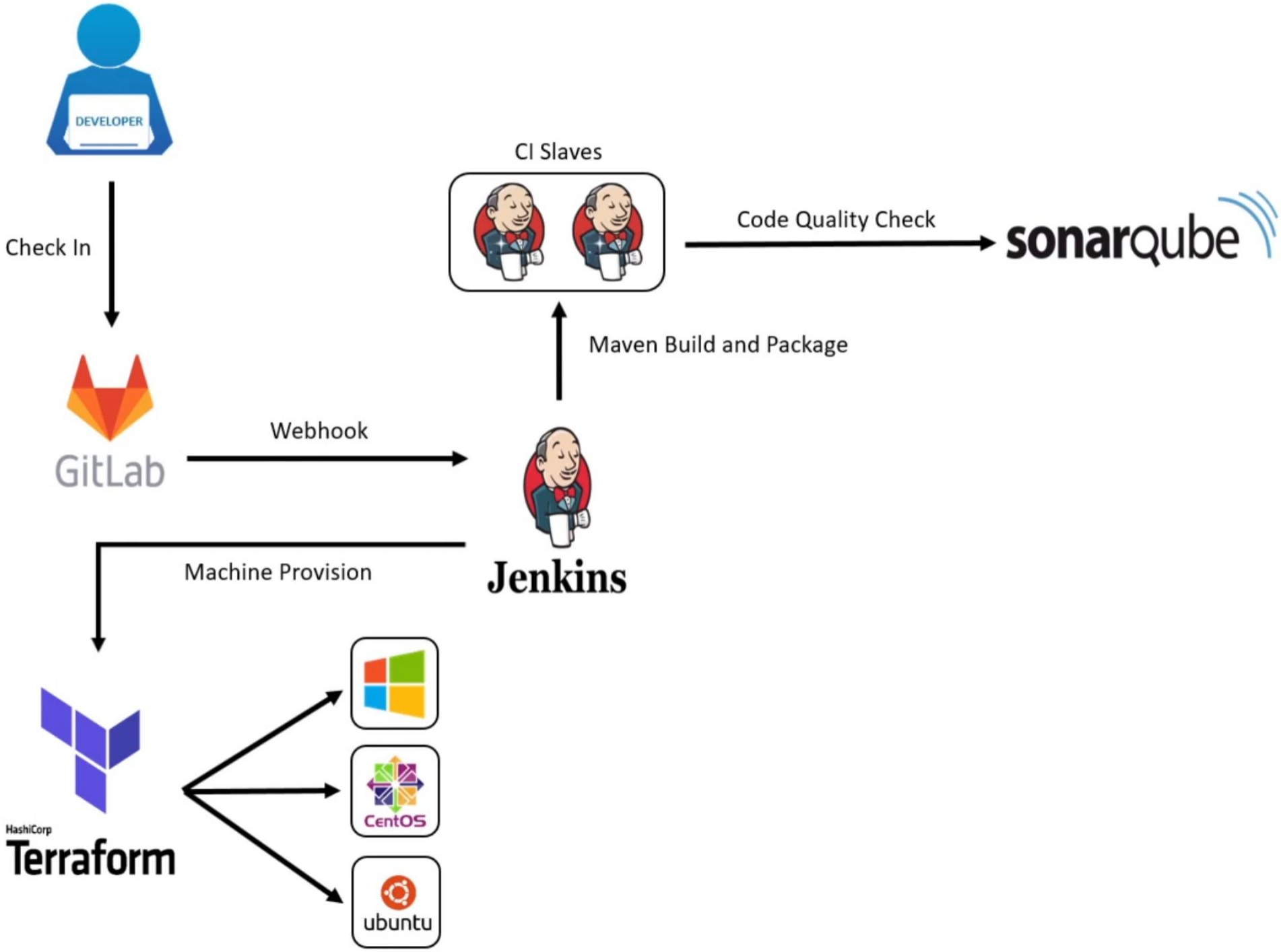
Jenkins

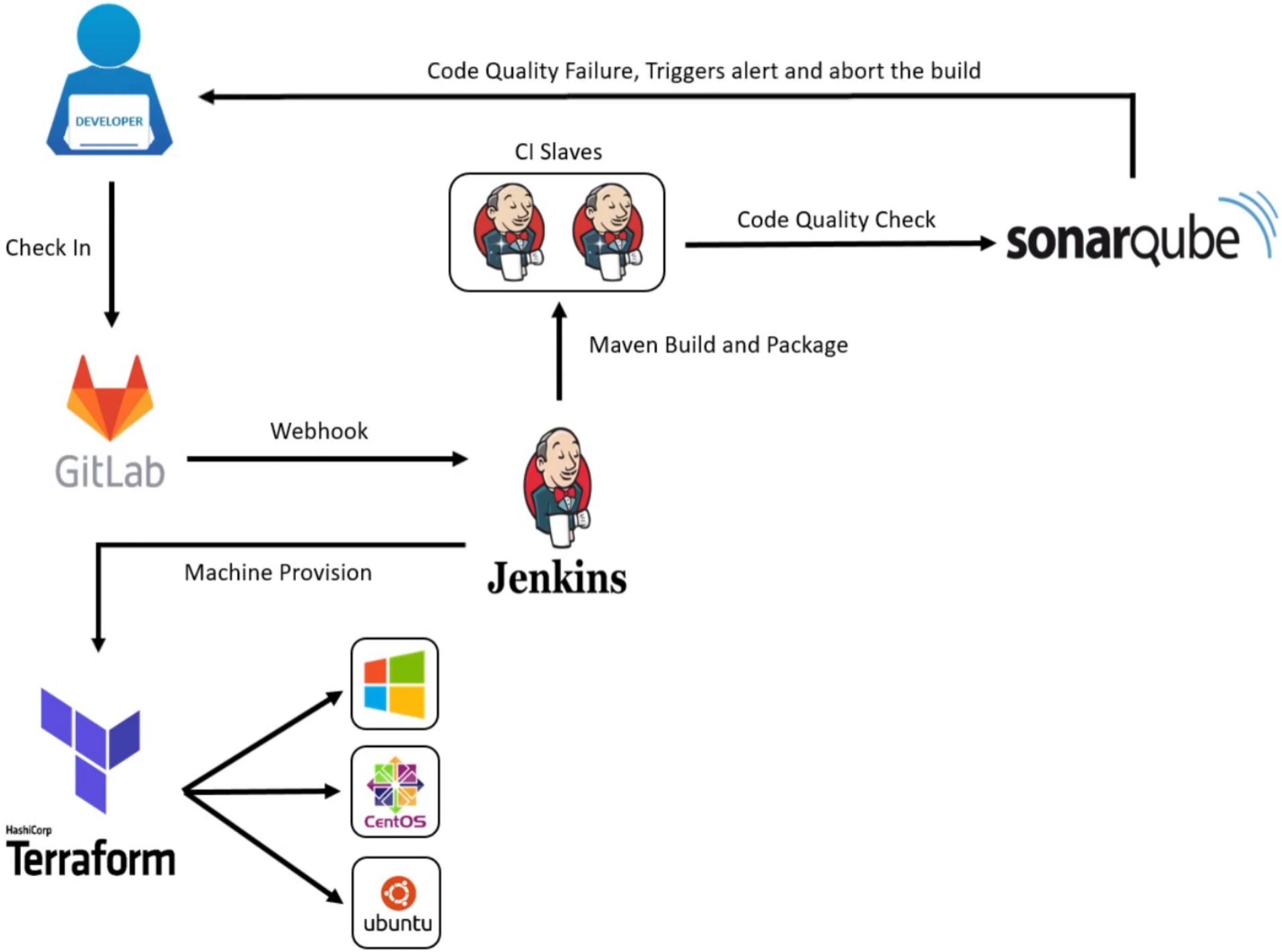

Machine Provision
Jenkins
HashiCorp
Terraform
CentOS
ubuntu

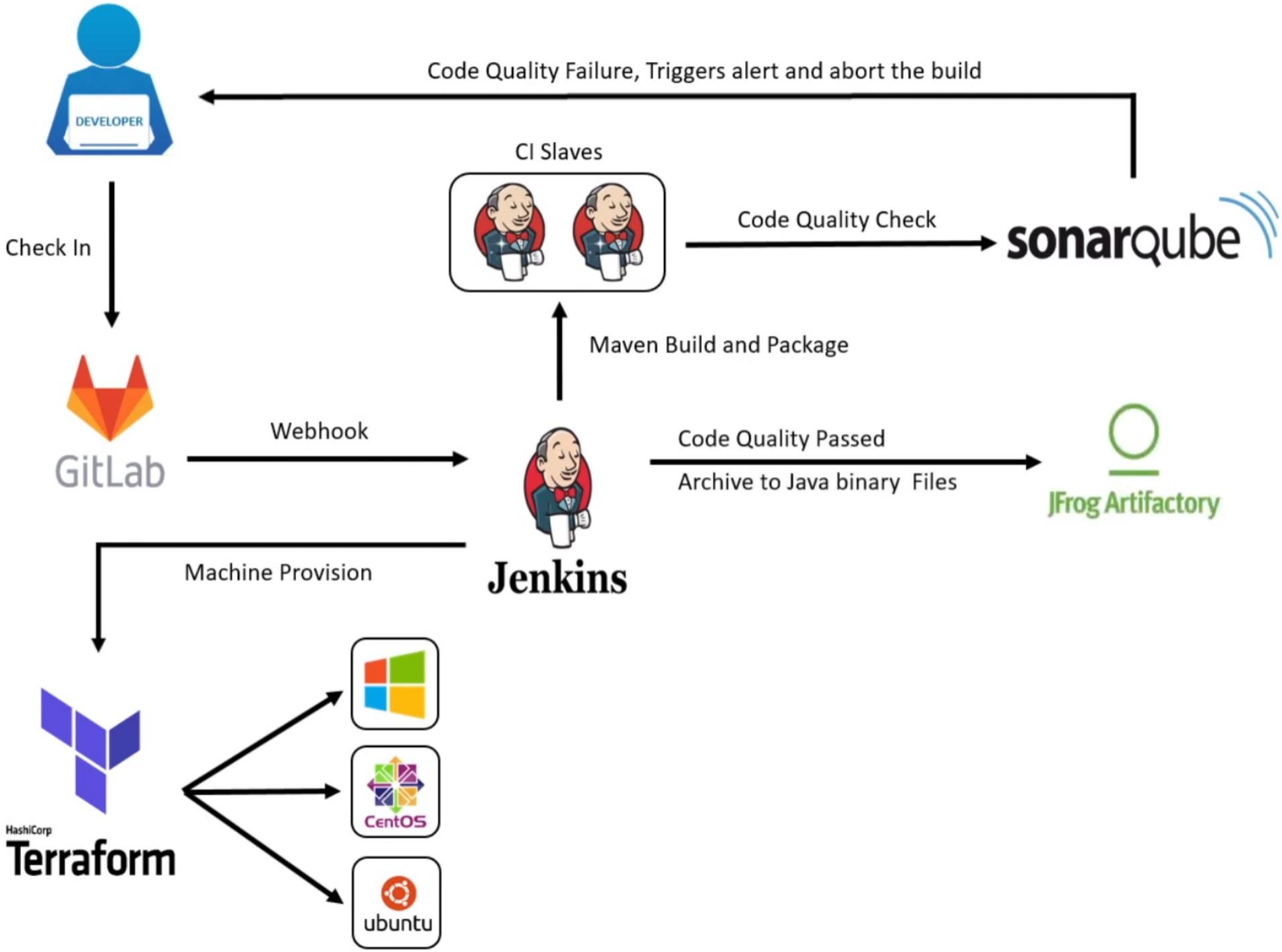

DEVELOPER
Check In
GitLab
Machine Provision
Jenkins
HashiCorp
Terraform
CentOS
ubuntu

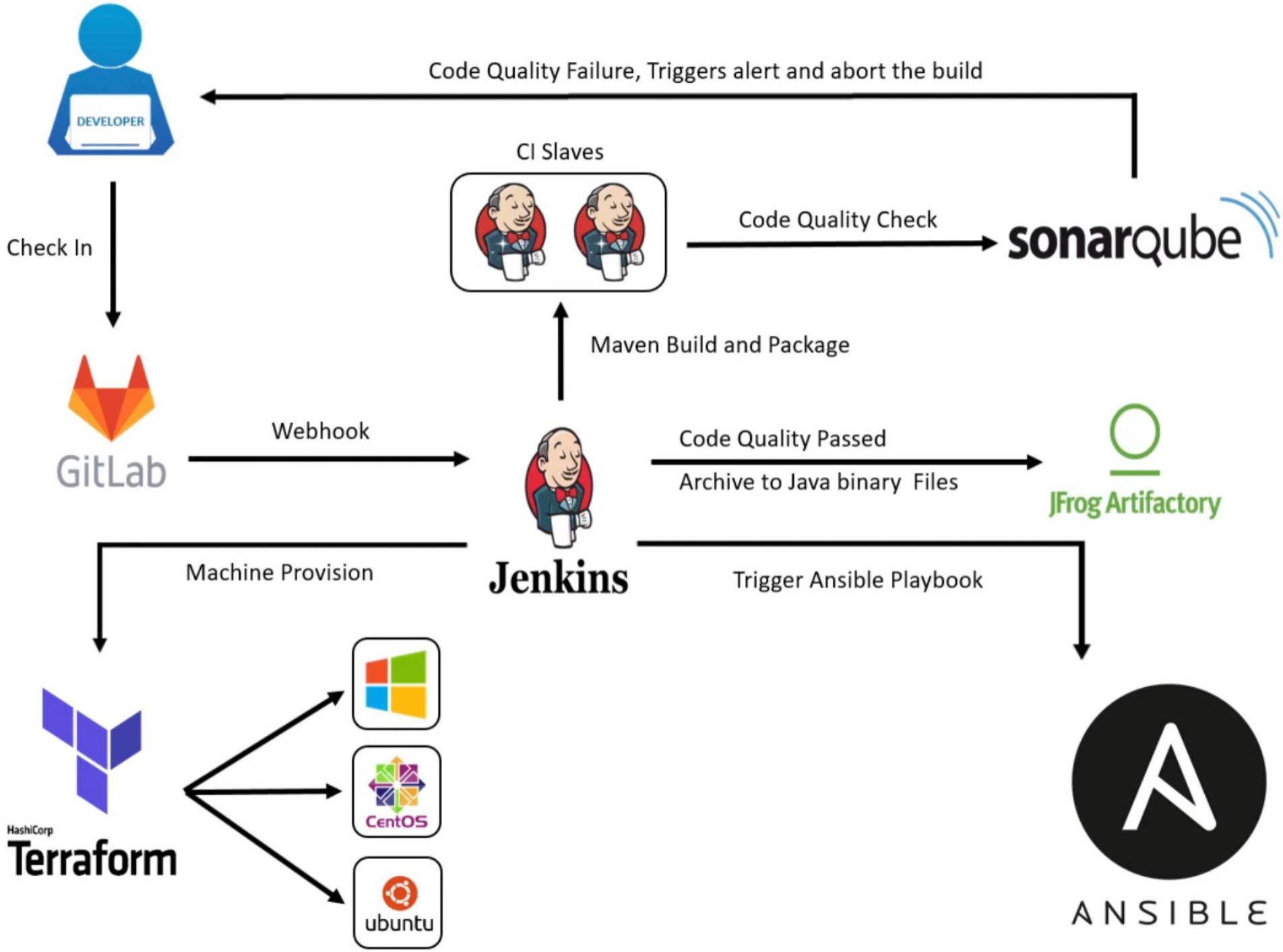

DEVELOPER
Check In
GitLab
Webhook
Jenkins
Machine Provision
HashiCorp
Terraform
CentOS
ubuntu

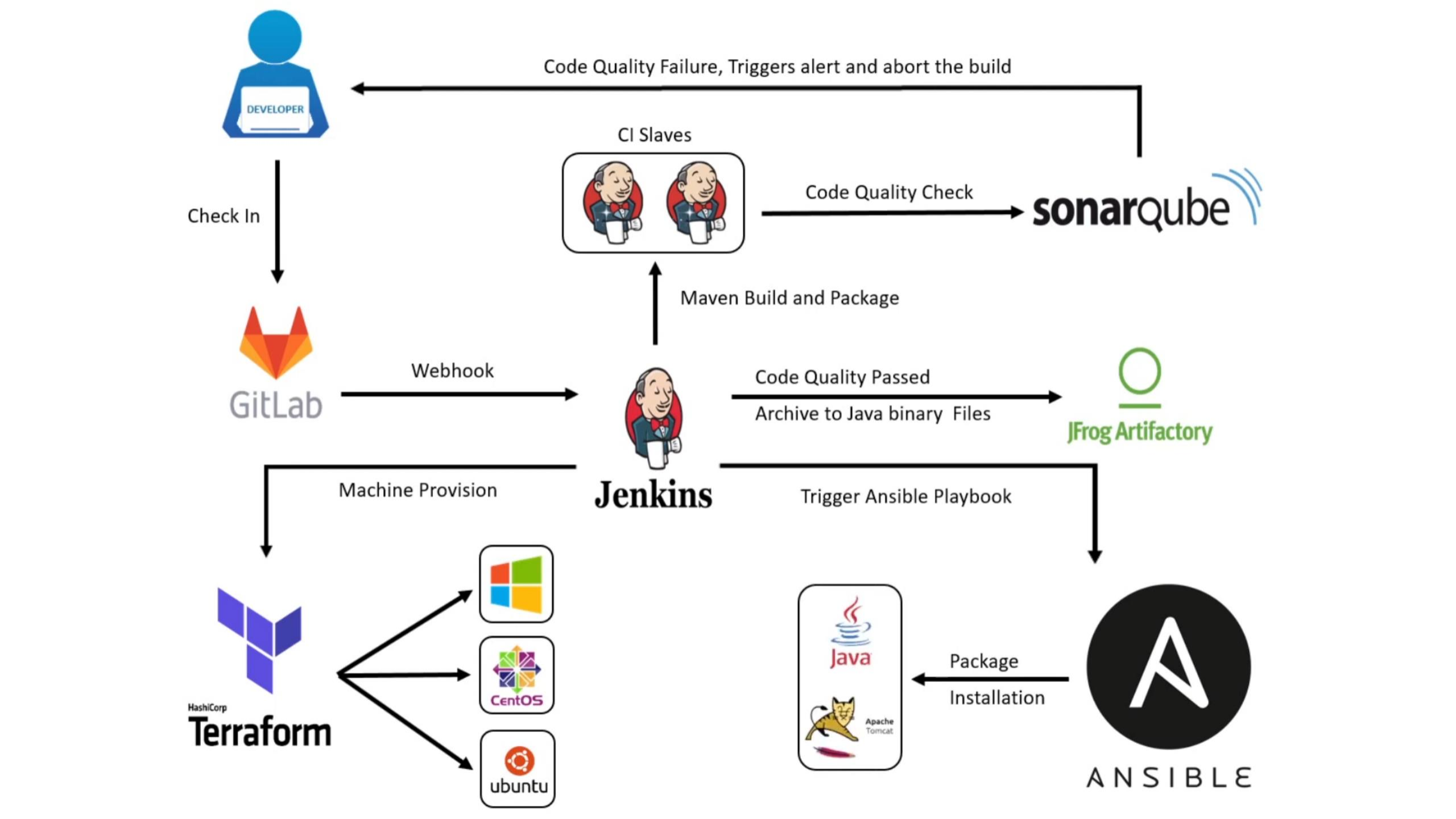

DEVELOPER
Check In
GitLab
Webhook
CI Slaves
Maven Build and Package
Jenkins
Machine Provision
HashiCorp
Terraform
CentOS
ubuntu

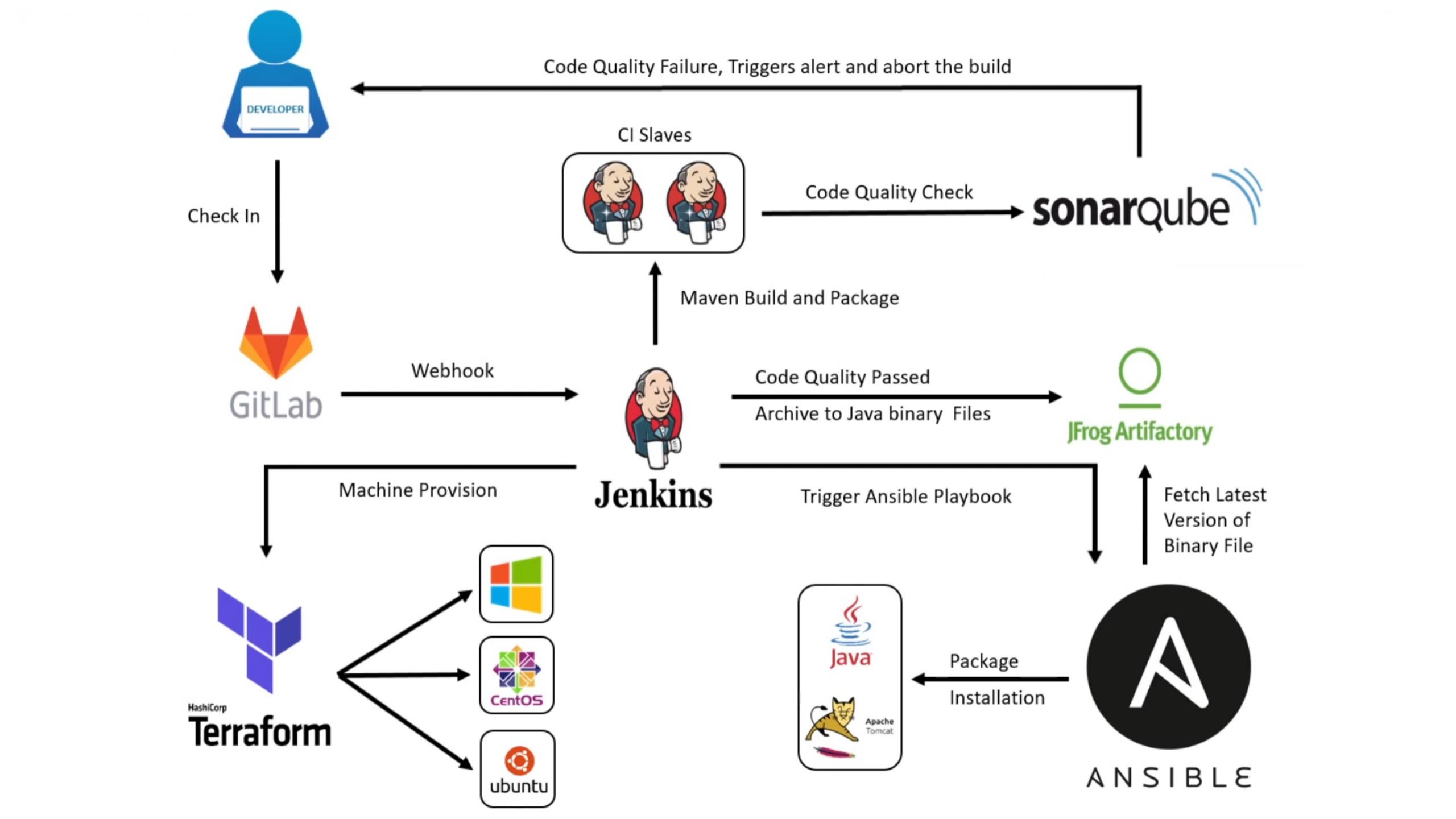

DEVELOPER
Check In
GitLab
Webhook
Jenkins
CI Slaves
Code Quality Check
sonarqube
Maven Build and Package
Machine Provision
HashiCorp
Terraform
CentOS
ubuntu

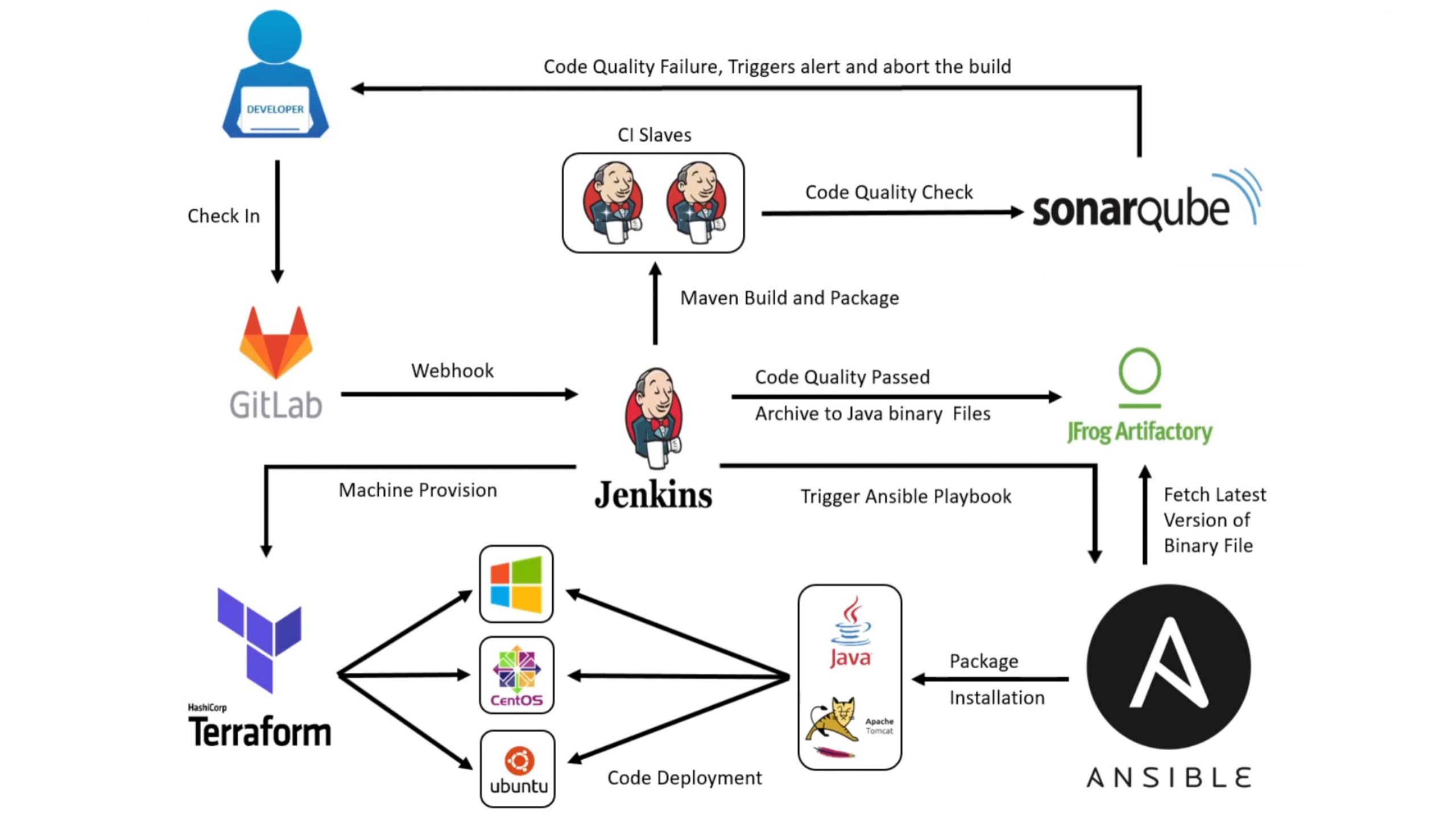

DEVELOPER
Code Quality Failure, Triggers alert and abort the build
CI Slaves
Check In
Code Quality Check
sonarqube
Maven Build and Package
GitLab
Webhook
Jenkins
Machine Provision
HashiCorp
Terraform
CentOS
ubuntu

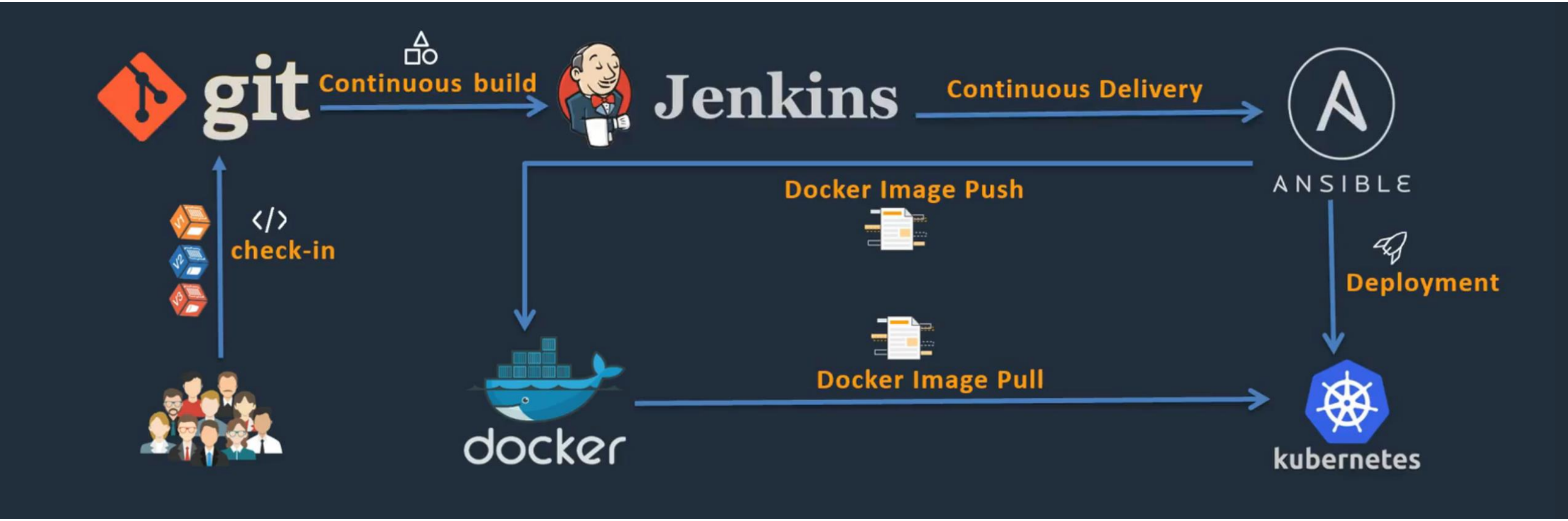

DEVELOPER
Code Quality Failure, Triggers alert and abort the build
CI Slaves
Check In
Code Quality Check
sonarqube
Maven Build and Package
GitLab
Webhook
Code Quality Passed
Archive to Java binary Files
JFrog Artifactory
Jenkins
Machine Provision
HashiCorp
Terraform
CentOS
ubuntu


DEVELOPER
Code Quality Failure, Triggers alert and abort the build
CI Slaves
Check In
Code Quality Check
sonarqube
Maven Build and Package
GitLab
Webhook
Code Quality Passed
Archive to Java binary Files
JFrog Artifactory
Machine Provision
Jenkins
Trigger Ansible Playbook
HashiCorp
Terraform
CentOS
ubuntu
ANSIBLE


DEVELOPER
Code Quality Failure, Triggers alert and abort the build
CI Slaves
Check In
Code Quality Check
sonarqube
Maven Build and Package
GitLab
Webhook
Code Quality Passed
Archive to Java binary Files
JFrog Artifactory
Jenkins
Machine Provision
Trigger Ansible Playbook
HashiCorp
Terraform
CentOS
ubuntu
Java
Apache
Tomcat
Package
Installation
ANSIBLE


DEVELOPER
Code Quality Failure, Triggers alert and abort the build
CI Slaves
Check In
Code Quality Check
sonarqube
Maven Build and Package
GitLab
Webhook
Code Quality Passed
Archive to Java binary Files
JFrog Artifactory
Jenkins
Machine Provision
Trigger Ansible Playbook
Fetch Latest Version of Binary File
HashiCorp Terraform
CentOS
ubuntu
Java
Apache Tomcat
Package Installation
ANSIBLE


DEVELOPER
Code Quality Failure, Triggers alert and abort the build
CI Slaves
Check In
Code Quality Check
sonarqube
Maven Build and Package
GitLab
Webhook
Code Quality Passed
Archive to Java binary Files
JFrog Artifactory
Jenkins
Machine Provision
Trigger Ansible Playbook
Fetch Latest Version of Binary File
HashiCorp Terraform
CentOS
ubuntu
Java
Apache Tomcat
Package Installation
ANSIBLE
Code Deployment


git
Continuous build
Jenkins
Continuous Delivery
ANSIBLE
Docker Image Push
check-in
Deployment
Docker Image Pull
docker
kubernetes

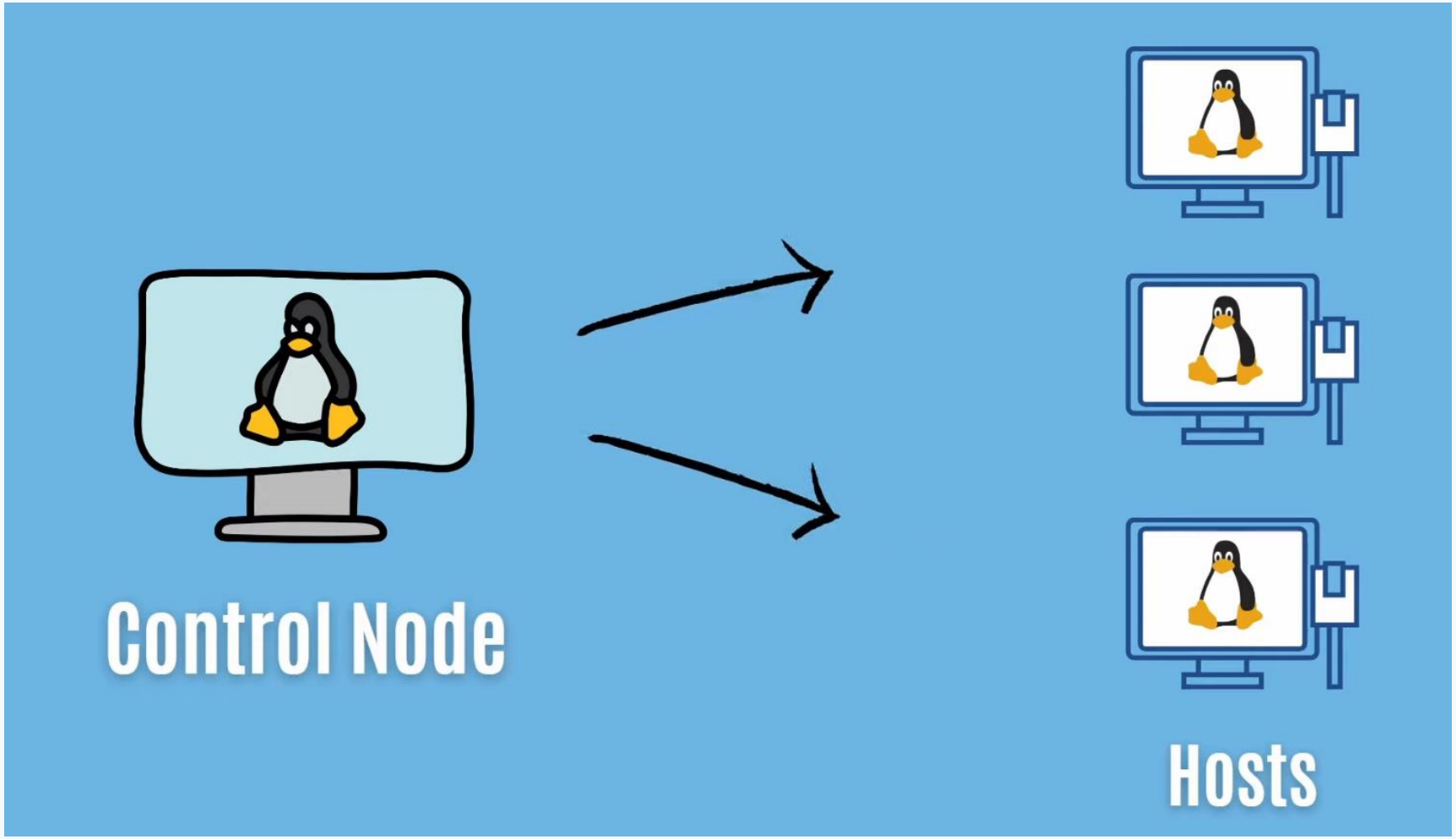
Limiting Playbook / Task Runs

- When writing Ansible, sometimes it is tedious to make a change in a playbook or task, then run the playbook It can sometimes be very helpful to run a module directly as shown above, but only against a single development host.
- Limit to one host
  - ansible-playbook playbooks/PLAYBOOK_NAME.yml --limit "host1“
- Limit to multiple hosts
  - ansible-playbook playbooks/PLAYBOOK_NAME.yml --limit "host1,host2“
- Negating
  - ansible-playbook playbooks/PLAYBOOK_NAME.yml --limit 'all:!host1'

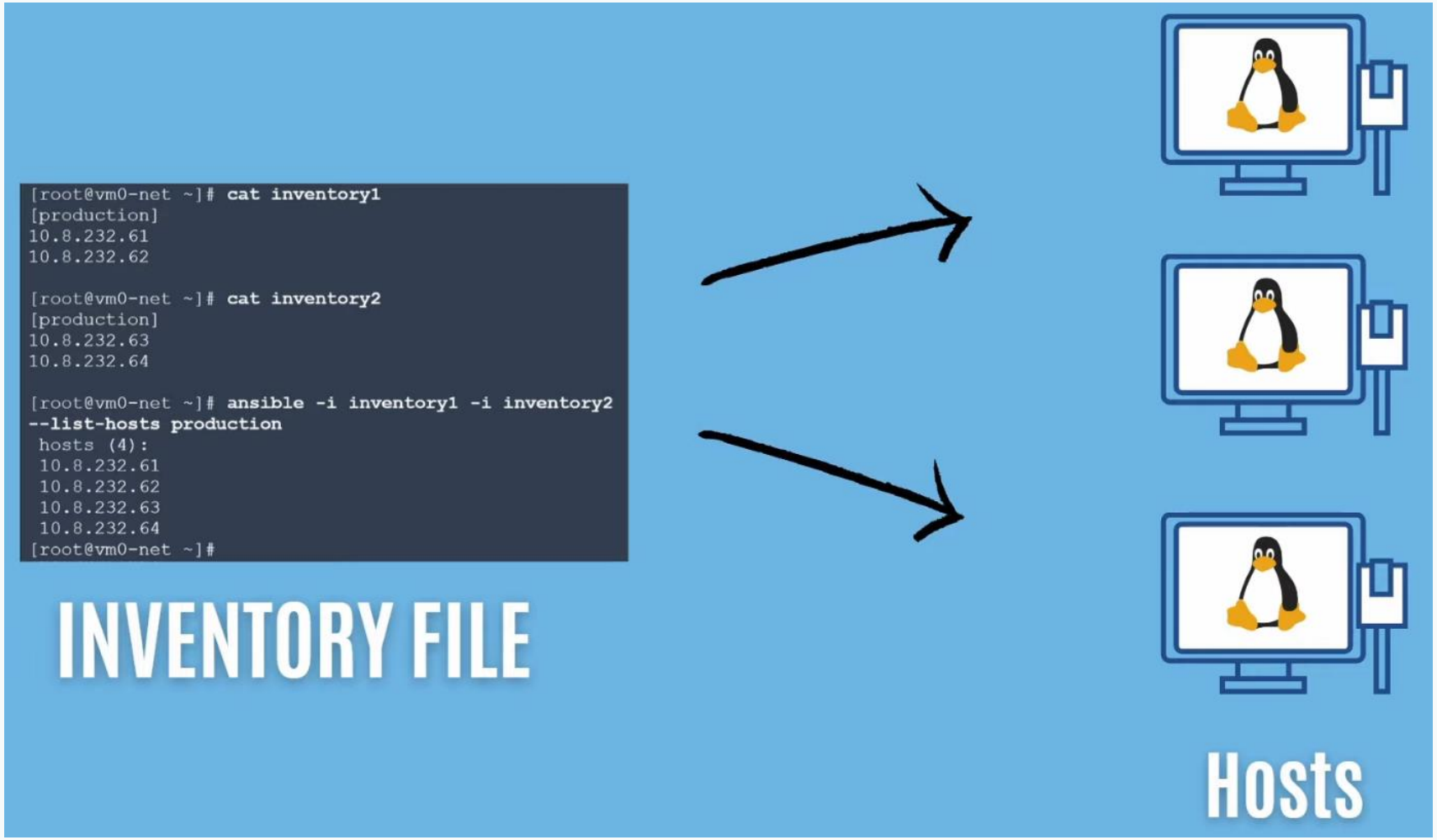
Limiting Playbook / Task Runs

- Limit to host group
  - ansible-playbook playbooks/PLAYBOOK_NAME.yml --limit 'group1’
- Limit to host group
  - ansible-playbook playbooks/PLAYBOOK_NAME.yml --limit 'group1’
- Limit to all tags matching install
  - ansible-playbook playbooks/PLAYBOOK_NAME.yml --tags 'install’
- Skip any tag matching sudoers
  - ansible-playbook playbooks/PLAYBOOK_NAME.yml --skip-tags 'sudoers'

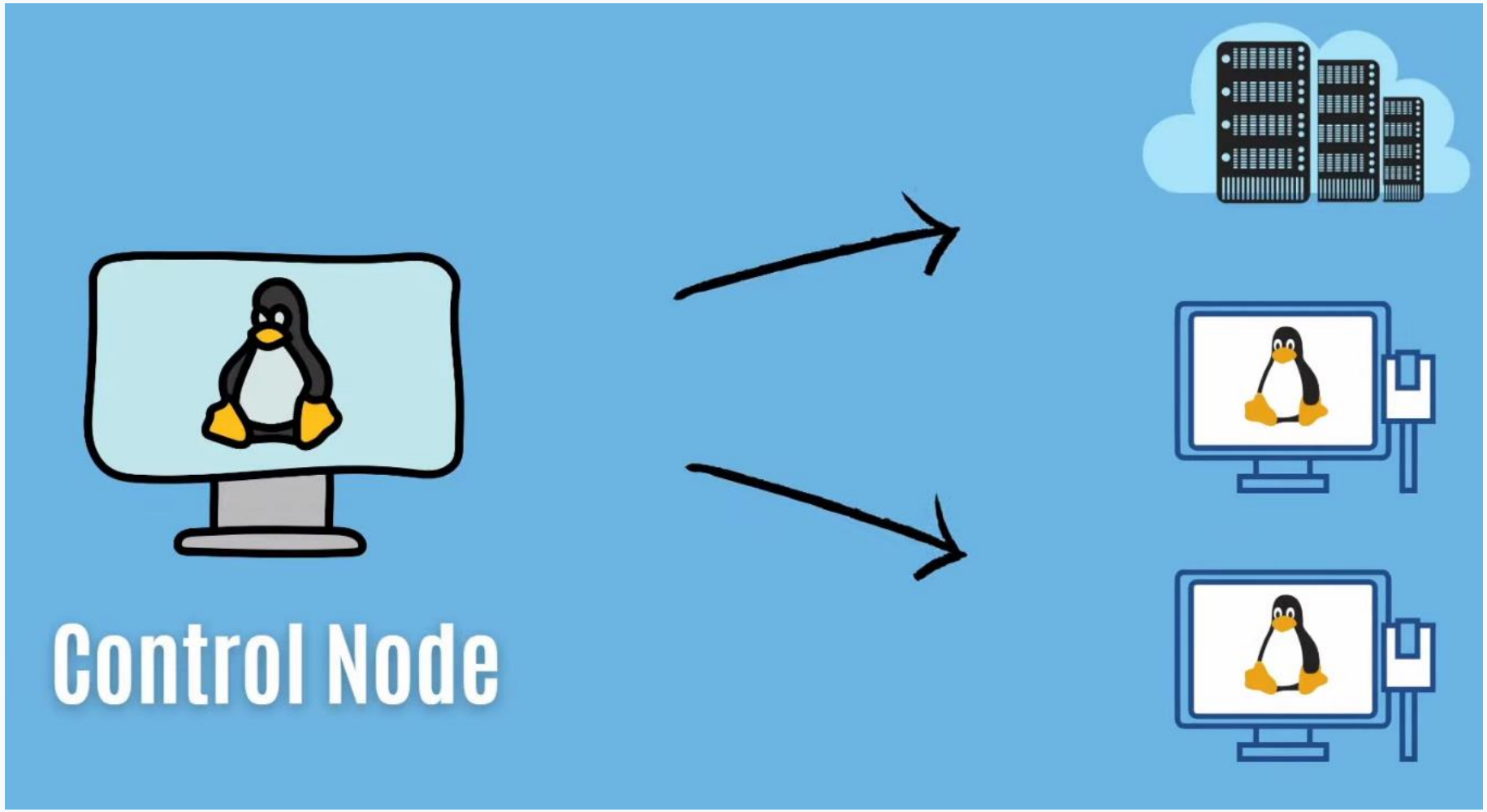
Dynamic Inventory



Dynamic Inventory



Dynamic Inventory



IAM and admin
IAM
Identity and organisation
Policy troubleshooter
Policy analyser
Organisation policies
Service accounts
Workload Identity Federat...
Labels
Tags
Settings
Privacy and security
Identity-Aware Proxy
Roles
Audit logs
Asset inventory
Manage resources
Release notes

Create service account

HELP ASSISTANT

1 Service account details

Service account name
ansible_admin

Display name for this service account

Service account ID *
ansible-admin

Email address: ansible-admin@venkata-practice.iam.gserviceaccount.com

Service account description
Ansible Admin

Describe what this service account will do

CREATE AND CONTINUE

2 Grant this service account access to the project (optional)

3 Grant users access to this service account (optional)

DONE CANCEL

IAM and admin
IAM
Identity and organisation
Policy troubleshooter
Policy analyser
Organisation policies
Service accounts
Workload Identity Federat...
Labels
Tags
Settings
Privacy and security
Identity-Aware Proxy
Roles
Audit logs
Asset inventory
Manage resources
Release notes

Create service account

HELP ASSISTANT

Service account details

2 Grant this service account access to the project (optional)

Grant this service account access to venkata-practice so that it has permission to complete specific actions on the resources in your project. Learn more

Role Condition

Filter Type to filter

Quick access
- Currently used
- Basic

By product or service
- Access Approval
- Access Context Manager
- Actions

Roles
- Cloud Functions Admin
- Compute Engine Service Agent
- Editor
- Owner

MANAGE ROLES

resources.

+ ADD ANOTHER ROLE

CONTINUE

Create service account

HELP ASSISTANT | HIDE INFO PANEL

Service account details

Grant this service account access to the project (optional)

3 Grant users access to this service account (optional)

Grant access to users or groups that need to perform actions as this service account. Learn more

Service account users role

Grant users the permissions to deploy jobs and VMs with this service account

Service account admins role

Grant users the permission to administer this service account

DONE CANCEL

Permissions

Show inherited permissions

Filter Enter property name or value

Role/Principal	Inheritance
Compute Engine Service Agent (1)	
Editor (2)	
Owner (2)	

IAM and admin

IAM
Identity and organisation
Policy troubleshooter
Policy analyser
Organisation policies
Service accounts
Workload Identity Federat...
Labels
Tags
Settings
Privacy and security
Identity-Aware Proxy
Roles
Audit logs
Asset inventory

Manage resources

Release notes

Service accounts

+ CREATE SERVICE ACCOUNT | DELETE | MANAGE ACCESS | REFRESH | HELP ASSISTANT | LEARN

Service accounts for project 'venkata-practice'

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps or systems running outside Google. Learn more about service accounts.

Organisation policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload or the creation of service accounts entirely. Learn more about service account organisation policies.

Filter Enter property name or value

Email	Status	Name	Description	Key ID	Key creation date	OAuth 2 client ID	Actions
ansible-admin@venkata-practice.iam.gserviceaccount.com	✓	ansible_admin	Ansible Admin	No keys		116404704201935240746	
660520246477-compute@developer.gserviceaccount.com	✓	Compute Engine default service account		No keys		113627740013025522152	
my-ansible@venkata-practice.iam.gserviceaccount.com	✓	my-ansible		No keys		105749895159505180992	

IAM and admin

IAM
Identity and organisation
Policy troubleshooter
Policy analyser
Organisation policies
Service accounts
Workload Identity Federat...
Labels
Tags
Settings
Privacy and security
Identity-Aware Proxy
Roles
Audit logs
Asset inventory

Manage resources

Release notes

ansible_admin

HELP ASSISTANT

DETAILS | PERMISSIONS | KEYS | METRICS | LOGS

Service account details

Name
ansible_admin SAVE

Description
Ansible Admin SAVE

Email
ansible-admin@venkata-practice.iam.gserviceaccount.com

Unique ID
116404704201935240746

Service account status

Disabling your account allows you to preserve your policies without having to delete it.

Account currently active

DISABLE SERVICE ACCOUNT

Advanced settings

Inbox (21) - cloud | ansible_admin – I | Roles – IAM and a | Mail - Venkata Su | Join conversation | Installing Ansible | Ansible Galaxy | Install Docker Eng | Install Docker Eng

console.cloud.google.com/iam-admin/serviceaccounts/details/116404704201935240746/keys?project=venkata-practice

Google Cloud | venkata-practice | Search Products, resources, docs (/)

IAM and admin

- IAM
- Identity and organisation
- Policy troubleshooter
- Policy analyser
- Organisation policies
- Service accounts
- Workload Identity Federat...
- Labels
- Tags
- Settings
- Privacy and security
- Identity-Aware Proxy
- Roles
- Audit logs
- Asset inventory
- Manage resources
- Release notes

ansible_admin

HELP ASSISTANT

DETAILS | PERMISSIONS | KEYS | METRICS | LOGS

Keys

Service account keys could pose a security risk if compromised. We recommend that you avoid downloading service account keys and instead use the Workload Identity Federation . You can learn more about the best way to authenticate service accounts on Google Cloud here .

Add a new key pair or upload a public key certificate from an existing key pair.

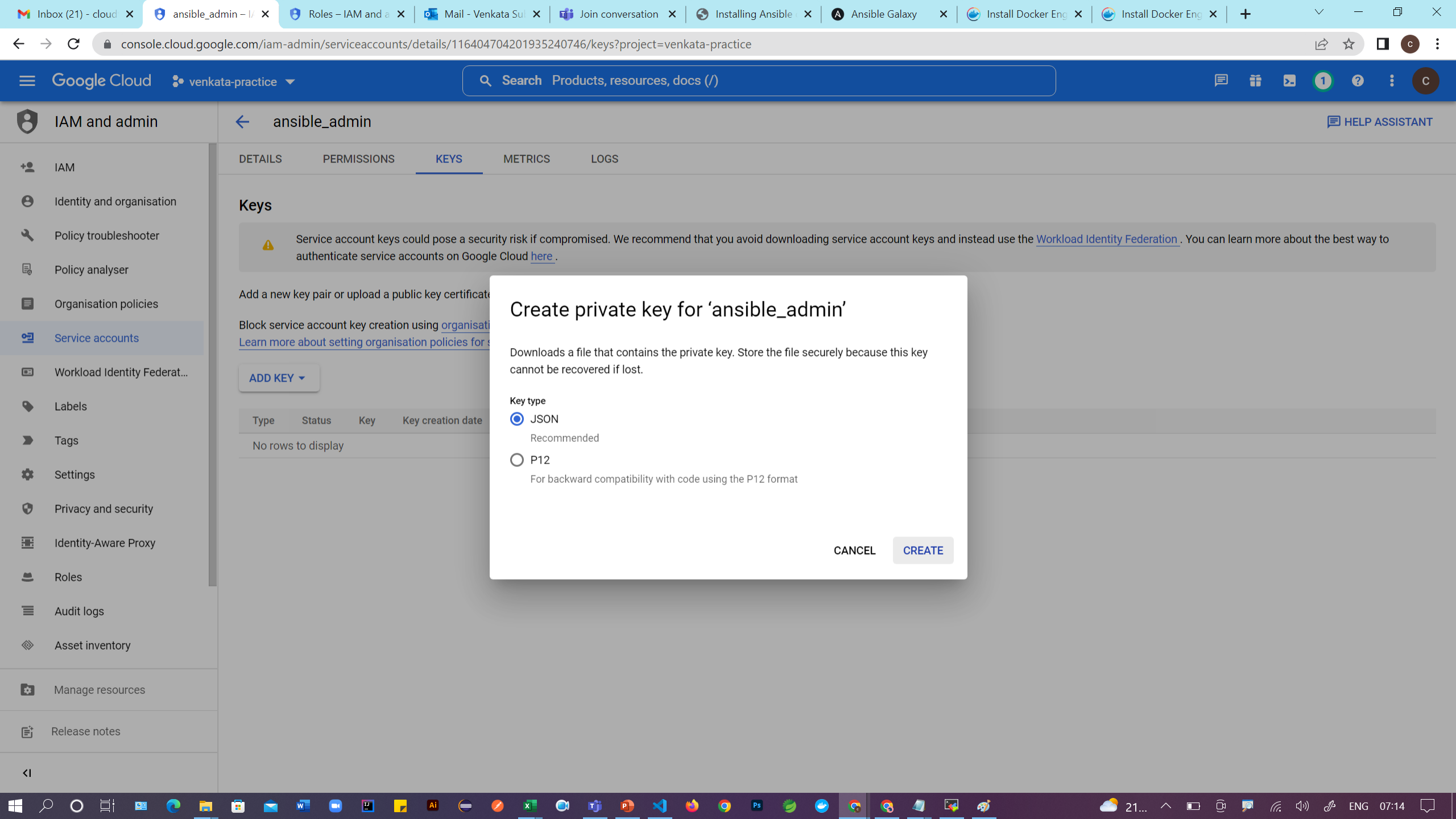
Block service account key creation using organisation policies.
Learn more about setting organisation policies for service accounts

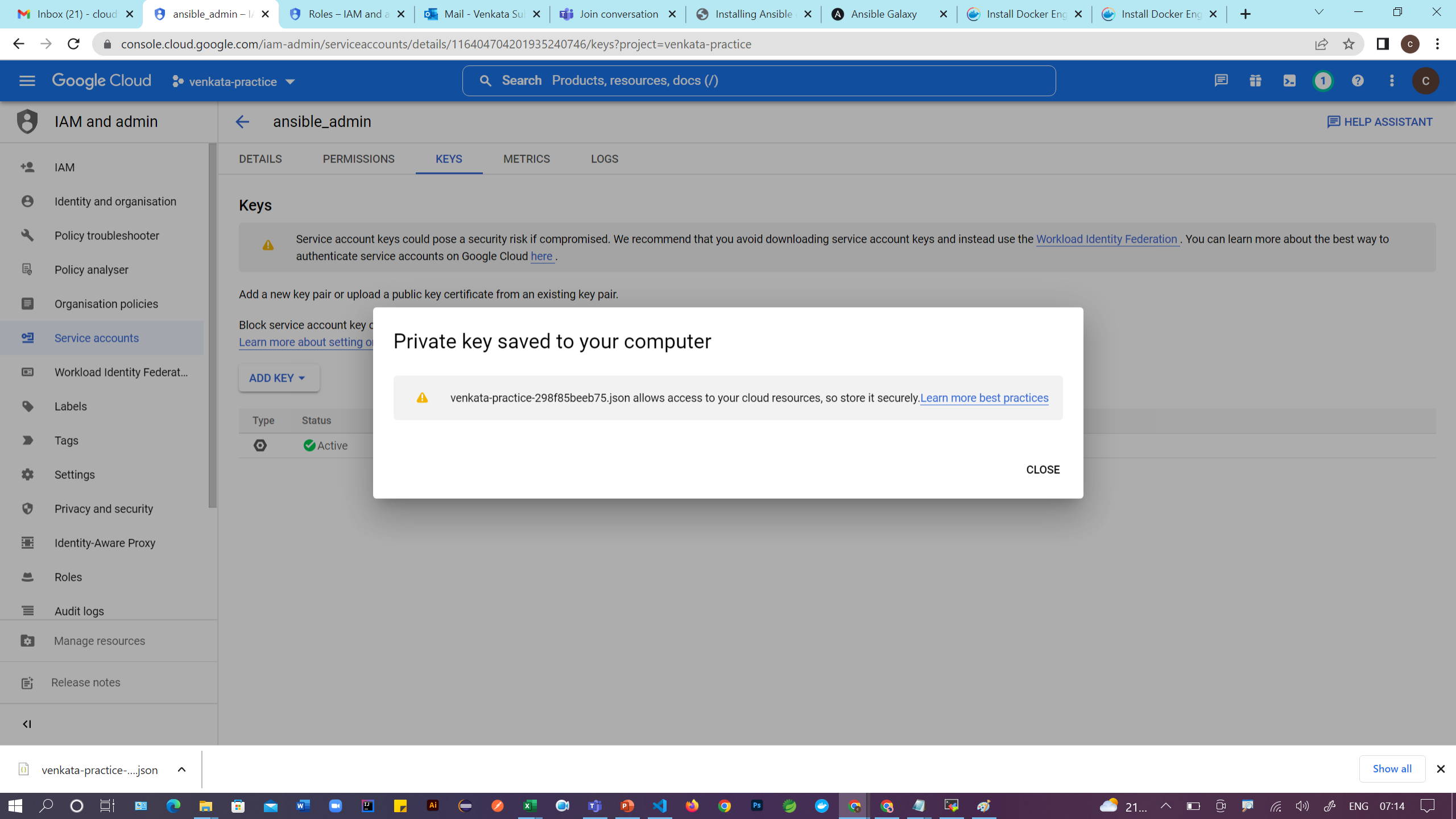
ADD KEY

Type	Status	Key	Key creation date	Key expiry date
No rows to display				

https://console.cloud.google.com/iam-admin/serviceaccounts/details/116404704201935240746/keys?project=venkata-practice

ENG 07:12


Inbox (21) - cloud
ansible_admin – I
Roles – IAM and a
Mail - Venkata Su
Join conversation
Installing Ansible
Ansible Galaxy
Install Docker Eng
Install Docker Eng
console.cloud.google.com/iam-admin/serviceaccounts/details/116404704201935240746/keys?project=venkata-practice
Google Cloud
venkata-practice
Search Products, resources, docs (/)
IAM and admin
ansible_admin
HELP ASSISTANT
DETAILS
PERMISSIONS
KEYS
METRICS
LOGS
IAM
Identity and organisation
Policy troubleshooter
Policy analyser
Organisation policies
Service accounts
Workload Identity Federat...
Labels
Tags
Settings
Privacy and security
Identity-Aware Proxy
Roles
Audit logs
Asset inventory
Manage resources
Release notes
Keys
Service account keys could pose a security risk if compromised. We recommend that you avoid downloading service account keys and instead use the Workload Identity Federation . You can learn more about the best way to authenticate service accounts on Google Cloud here .
Add a new key pair or upload a public key certificate
Block service account key creation using organisati
Learn more about setting organisation policies for
ADD KEY
Type
Status
Key
Key creation date
No rows to display
Create private key for 'ansible_admin'
Downloads a file that contains the private key. Store the file securely because this key cannot be recovered if lost.
Key type
JSON
Recommended
P12
For backward compatibility with code using the P12 format
CANCEL
CREATE
ENG
07:14


Inbox (21) - cloud
ansible_admin – I
Roles – IAM and a
Mail - Venkata Su
Join conversation
Installing Ansible
Ansible Galaxy
Install Docker Eng
Install Docker Eng
console.cloud.google.com/iam-admin/serviceaccounts/details/116404704201935240746/keys?project=venkata-practice
Google Cloud
venkata-practice
Search Products, resources, docs (/)
IAM and admin
ansible_admin
HELP ASSISTANT
IAM
Identity and organisation
Policy troubleshooter
Policy analyser
Organisation policies
Service accounts
Workload Identity Federat...
Labels
Tags
Settings
Privacy and security
Identity-Aware Proxy
Roles
Audit logs
Manage resources
Release notes
DETAILS
PERMISSIONS
KEYS
METRICS
LOGS
Keys
Service account keys could pose a security risk if compromised. We recommend that you avoid downloading service account keys and instead use the Workload Identity Federation . You can learn more about the best way to authenticate service accounts on Google Cloud here .
Add a new key pair or upload a public key certificate from an existing key pair.
Block service account key c
Learn more about setting o
ADD KEY
Type
Status
Active
Private key saved to your computer
venkata-practice-298f85beeb75.json allows access to your cloud resources, so store it securely.Learn more best practices
CLOSE
venkata-practice-....json
Show all
ENG
07:14

Optimizing Playbooks

Identify slow tasks with callback plugins

A specific task in a playbook might look simple, but it can be why the playbook is executing slowly. You can enable callback plugins such as timer, profile_tasks, and profile_roles to find a task's time consumption and identify which jobs are slowing down your plays.

Configure ansible.cfg with the plugins:

```
[defaults]
inventory = ./hosts
callbacks_enabled = timer, profile_tasks, profile_roles
```

Now execute the ansible-playbook command:

```
ansible-playbook yourplaybook.yml
```

https://www.ansible.com/resources/ebooks/mastering-ansible?intcmp=701f20000012ngPAAQ

Optimizing Playbooks

Disable fact gathering

When a playbook executes, each play runs a hidden task, called gathering facts, using the setup module. This gathers information about the remote node you're automating, and the details are available under the variable ansible_facts. But if you're not using these details in your playbook anywhere, then this is a waste of time. You can disable this operation by setting gather_facts: False in the play.

With gathering facts enabled **performance decreases**

With gather_facts: False disabling fact gathering, **performance increases**

Optimizing Playbooks

Configure parallelism:

Ansible uses batches for task execution, which are controlled by a parameter called forks. The default value for forks is 5, which means Ansible executes a task on the first five hosts, waits for the task to complete, and then takes the next batch of five hosts, and so on. Once all hosts finish the task, Ansible moves to the next tasks with a batch of five hosts again.

You can increase the value of forks in ansible.cfg, enabling Ansible to execute a task on more hosts in parallel:

```
[defaults]
inventory = ./hosts
forks=50
```

You can also change the value of forks dynamically while executing a playbook by using the --forks option (-f for short):

```
ansible-playbook site.yaml --forks 50
```

A word of warning: When Ansible works on multiple managed nodes, it uses more computing resources (CPU and memory). Based on your Ansible control node machine capacity, configure forks appropriately and responsibly.

Optimizing Playbooks

Configure SSH optimization:

Establishing a secure shell (SSH) connection is a relatively slow process that runs in the background. The global execution time increases significantly when you have more tasks in a playbook and more managed nodes to execute the tasks.

You can use **ControlMaster** and **ControlPersist** features in **ansible.cfg** (in the ssh_connection section) to mitigate this issue.

ControlMaster allows multiple simultaneous SSH sessions with a remote host to use a single network connection. This saves time on an SSH connection's initial processes because later SSH sessions use the first SSH connection for task execution.

ControlPersist indicates how long the SSH keeps an idle connection open in the background. For example, ControlPersist=60s keeps the connection idle for 60 seconds:

```
[ssh_connection]
ssh_args = -o ControlMaster=auto -o ControlPersist=60s
```

Optimizing Playbooks

Disable host key checking in a dynamic environment:

By default, Ansible checks and verifies SSH host keys to safeguard against server spoofing and man-in-the-middle attacks. This also consumes time. If your environment contains immutable managed nodes (virtual machines or containers), then the key is different when the host is reinstalled or recreated. You can disable host key checking for such environments by adding the host_key_checking parameter in your ansible.cfg file and setting it to False:

```
[defaults]
host_key_checking = False
```

I don't recommend this outside of a controlled environment. Make sure you have a clear understanding of the implications of this action before you use it in critical environments.

Optimizing Playbooks

Use pipelining:

When Ansible uses SSH, several SSH operations happen in the background for copying files, scripts, and other execution commands. You can reduce the number of SSH connections by enabling the pipelining parameter (it's disabled by default) in ansible.cfg:

```
# ansible.cfg
            pipelining = True
```

Optimizing Playbooks

Use execution strategies

By default, Ansible waits for every host to finish a task before moving to the next task, which is called linear strategy.

If you don't have dependencies on tasks or managed nodes, you can change strategy to free, which allows Ansible to execute tasks on managed hosts until the end of the play without waiting for other hosts to finish their tasks:

```
- hosts: production servers
  strategy: free
  tasks:
```

You can develop or use more strategy plugins as needed, such as Mitogen, which uses Python-based executions and connections.

Optimizing Playbooks

Use async tasks:

When a task executes, Ansible waits for it to complete before closing the connection to the managed node. This can become a bottleneck when you have tasks with longer execution times (such as disk backups, package installation, and so on) because it increases global execution time. If the following tasks do not depend on this long-running task, you can use the async mode with an appropriate poll interval to tell Ansible not to wait and proceed with the next tasks:

```
---
- name: Async Demo
  hosts: nodes
  tasks:

    - name: Initiate custom snapshot
      shell:
        "/opt/diskutils/snapshot.sh init"
      async: 120 # Maximum allowed time in Seconds
      poll: 05 # Polling Interval in Seconds
```

Questions ???